Linear Regression analysis

Silvia Saloni

Monday, July 25, 2016

Introduction

The aim of the project was to be able to find a model that best predict the relationship between: * the number of people counted with the survey for a given class at a particular hour * Wi-fi log counted in that room at that hour

This will allow to see whether Wi-Fi log is a good predictor for estimating occupancy in a classroom.

We first tried to see if the relationship between this two variables was linear. To do so we run a linear regression.

Below we describes step by step all the analysis performed.

```
## Loading required package: DBI
```

First of all, we set up the connection to the database, using the following code:

```
connection <- dbConnect(MySQL(),user="root", password="",dbname="mysql", host="localhost")
```

Then we made a query to the database, in order to get all the groundth truth data collected in room B.002, B.004 and B.006 from 9 to 17 and the correspondent Wi-Fi Log measured in that time frame and rooms.

The dataset created had in total 216 rows and it will allow us to explore if Wi-Fi log is a good predictor of the observed occupancy of the room in that particular hour.

As a target features for our linear regression we decided to use the number of associated client, calculated multiplying the percentage of the room full with the capacity of the room.

As response variables or feature we considered Wi-Fi logs, which were summarised either as average of the logs counted for each room and for each hour or as maximum of the logs measured for each room and for each hour.

Together with the Wi-Fi log, we included in the data set the following features:

- Date, which we did not use in this analysis, because they just cover 2 weeks of Novemeber, but for future analyses they can be used to group observations by seasons or semesters or to finds seasonal trends for time series analyses.
- Time, which will be explored either as continous variable and as categorical to explore if the time of the day can have an affect on the Wi-Fi log. To do so we, bin the time in 4 ranges: early morning (9-11), late morning (11-13), early afternoon (13-15) and late afternoon (15-17).

This will allow us to see if the Wi-Fi log accuracy was changing during the day. For example, it is more likely that early morning all the elctronic devices are fully powered and consequently the Wi-fi log data can be more accurate or overestimating the occupancy of the room (i.e. more than one device per person). On the contrary in the afternoon, the devices may be more likely to be out of battery and it is possible that there are less devices in the room.

- Module, which we are not going to include in the analysis because the majority of the module present are for computer science, but for future analyses it will be possible to explore if the Wi-fi log accuracy in predicting the occupancy change acroos the courses. Science course or computer science course will more likely to use electronic devices during lecture than art students.
- Course level, which can indicate us whether electronic devices will be less used during different course level. For example, first and second level course can be less practical and therefore laptop are not needed and that can decrease the number of devices connected. On the other hand, undergraduate might be more distracted during lecture and look at their phones during lectures. This will result in an increase of connection in that hour.
- Tutorial, which can affect the number of logged people. First of all, because tutorial divided the room in 2 and therefore there will be measured less people than expected.
- Double_module, categorical variable indicating wether in the class there are more than one module, increasing the number of people expected in the room.
- Double_module, categorical variable indicating wether in the class went ahead to correct the .

The resulting data set is printed below:

```
head(AnalysisTable)
```

```
##   Room       Date Time    Module Course_Level Tutorial Double_module
## 1    1 2015-11-03    9         0            0        0             0
## 2    1 2015-11-04    9 COMP30190            3        0             0
## 3    1 2015-11-05    9         0            0        0             0
## 4    1 2015-11-06    9 COMP30220            3        0             0
## 5    1 2015-11-09    9 COMP30190            3        0             0
## 6    1 2015-11-10    9         0            0        0             0
##   Class_went_ahead Capacity Percentage_room_full Average_clients
## 1                1       90                 0.00          4.7500
## 2                1       90                 0.25         13.4545
## 3                1       90                 0.00          6.8333
## 4                1       90                 0.00          2.4167
## 5                1       90                 0.25         14.7273
## 6                1       90                 0.00          2.2727
##   Max_clients Counted_client   Factor_Time
## 1          21            0.0 Early Morning
## 2          15           22.5 Early Morning
## 3          29            0.0 Early Morning
## 4           3            0.0 Early Morning
## 5          18           22.5 Early Morning
## 6          14            0.0 Early Morning
```

DATA QUALITY REPORT

Before running any analyses, we carried out the data quality report to check for any issue related to the variable (e.g. outlier, skewed distribution, NaN values) and solutions we will implement to solve them.

Initially we printed the descriptive statistic for all the features and the presence of NaN values.

```
summary(AnalysisTable)
```

```
##       Room       Date                Time          Module
##  Min.   :1   Length:216         Min.   : 9.00   Length:216
##  1st Qu.:1   Class :character   1st Qu.:10.75   Class :character
##  Median :2   Mode  :character   Median :12.50   Mode  :character
##  Mean   :2                      Mean   :12.50
##  3rd Qu.:3                      3rd Qu.:14.25
##  Max.   :3                      Max.   :16.00
##  Course_Level    Tutorial       Double_module Class_went_ahead
##  0:63         Min.   :0.00000   0:210         0: 22
##  1:14         1st Qu.:0.00000   1:  6         1:194
##  2:23         Median :0.00000
##  3:76         Mean   :0.02778
##  4:40         3rd Qu.:0.00000
##               Max.   :1.00000
##     Capacity     Percentage_room_full Average_clients   Max_clients
##  Min.   : 90.0   Min.   :0.00         Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 90.0   1st Qu.:0.00         1st Qu.: 11.61   1st Qu.: 18.75
##  Median : 90.0   Median :0.25         Median : 23.67   Median : 32.50
##  Mean   :113.3   Mean   :0.25         Mean   : 30.33   Mean   : 40.01
##  3rd Qu.:160.0   3rd Qu.:0.25         3rd Qu.: 43.88   3rd Qu.: 55.25
##  Max.   :160.0   Max.   :1.00         Max.   :192.92   Max.   :230.00
##  Counted_client            Factor_Time
##  Min.   :  0.00   Early Morning  :54
##  1st Qu.:  0.00   Late Morning   :54
##  Median : 22.50   Early Afternoon:54
##  Mean   : 28.01   Late Afternoon :54
##  3rd Qu.: 40.00
##  Max.   :160.00
```

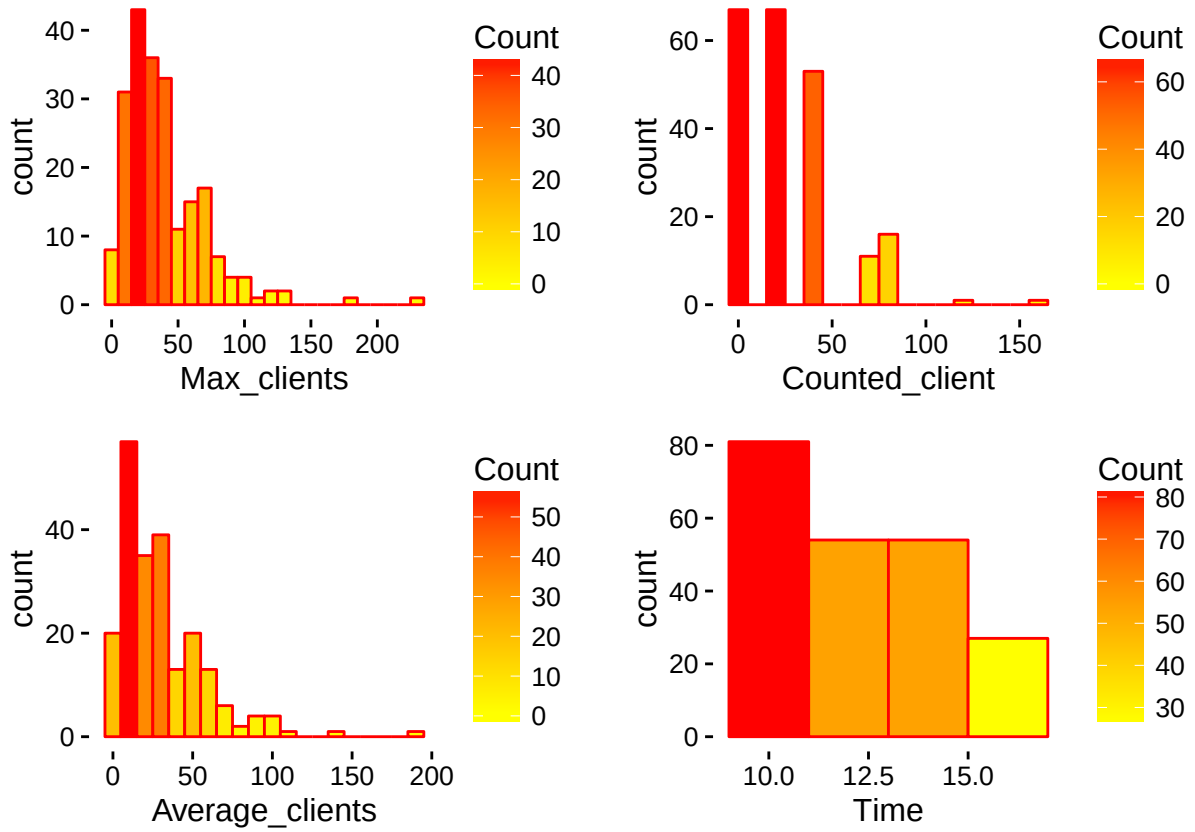
From this we could see that NaN values were not present in the data set. Furthermore, we could notice that the observations for the features Tutorials and Double_model were not even didtributed across the 2 levels of the variables. In fact, only 6 observation were present for tutorial class and for double module class. Therefore, we decided to discard both the features, because they will be not informative for the analysis. Similarly for the feature class_went_ahead, it results that the majority of the lectures did occur and we decided to discard it.

For the remaining variables we decided to plot:

- histograms and boxplots for continuous variables for exploring their distributions and check for outliers.
- bar plot for categorical features.

Histograms

```
## Loading required package: grid
```



Form the histograms we could see that the distribution of the feature Maximum_client (i.e. the Maximum number of devices logged in one hour lecture) was skewed to the left, indicating that the in the majority of the lecture were counted no more than 40 people. Similar was the case for the Average_client (i.e. average number of devices logged in one hour lecture), which indicated that in the class there were no more than 40 people. Different was the situation of the target feature, Counted client, which showed a skewed distribution, but more scattered, similar to a Poisson distribution. This can cause a problem in running a linear regression and more likely we have have to run a generalise linear model with a Poisson distribution. This is not surprising, since we are dealing with count data (Zuur et al. 2009). Feature times had as well a skewed distribution #make the boxplot for continuous variable box1 <- ggplot(AnalysisTable, aes(x = factor(0), y = Counted_client)) + geom_boxplot() + xlab("Expected students") + scale_x_discrete(breaks = NULL) + coord_flip() + theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

box2 <- ggplot(AnalysisTable, aes(x = factor(0), y = Average_clients)) + geom_boxplot() + xlab("Average counted students") + scale_x_discrete(breaks = NULL) + coord_flip() + theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

box3 <- ggplot(AnalysisTable, aes(x = factor(0), y =Max_clients)) + geom_boxplot() + xlab("Maximum counted students") + scale_x_discrete(breaks = NULL) + coord_flip() + theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

box4 <- ggplot(AnalysisTable, aes(x = factor(0), y = Time)) + geom_boxplot() + xlab("Maximum counted students") + scale_x_discrete(breaks = NULL) + coord_flip() + theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

plot all the histograms in one window

multiplot(box1, box2, box3, box4, cols=2)

GRAPH FOR CATEGORICAL DATA

bar1 <- ggplot(AnalysisTable, aes(x = Room)) + geom_bar(fill="seagreen4")+ theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

bar2 <- ggplot(AnalysisTable, aes(x = Course_Level)) + geom_bar(fill="seagreen4")+ theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

bar3 <- ggplot(AnalysisTable, aes(x = Binned_Percentage)) + geom_bar(fill="seagreen4")+ theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

bar4 <- ggplot(AnalysisTable, aes(x = Binned_Time)) + geom_bar(fill="seagreen4")+ theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

bar2 <- ggplot(AnalysisTable, aes(x = Module)) + geom_bar(fill="seagreen4

bar3 <- ggplot(AnalysisTable, aes(x = Tutorial)) + geom_bar(fill="seagreen4

bar4 <- ggplot(AnalysisTable, aes(x = Double_module)) + geom_bar(fill="seagreen4")

bar5 <- ggplot(AnalysisTable, aes(x = Class_went_ahead)) + geom_bar(fill="seagreen4

multiplot(bar1, bar2, bar3, bar4, cols=2)

RELATIONSHIP BETWEEN Variables

CASE 1: TARGET FEATURE = Counted_clients

–> Relationship with continous variables

Correlation Matrix

ggpairs(AnalysisTable, columns = c('Counted_client','Max_clients', 'Average_clients', 'Time'))+ geom_point() + geom_smooth(method=lm, fill="blue", color="blue", ...)

my_fn <- function(data, mapping, ...){ p <- ggplot(data = data, mapping = mapping) + geom_point() + geom_smooth(method=lm, fill="orangered3", color="orangered3", ...) p }

ggpairs(AnalysisTable, columns = c('Counted_client','Max_clients', 'Average_clients', 'Time'), lower = list(continuous = my_fn)) + theme_bw()

–> Relationship with categorical variables

Box plot

pairbox1 <- ggplot(AnalysisTable, aes(x = Room, y =Counted_client)) + geom_boxplot()+ theme_bw()+theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

pairbox2 <- ggplot(AnalysisTable, aes(x = Binned_Time , y = Counted_client)) + geom_boxplot() + theme_bw()+theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

pairbox3 <- ggplot(AnalysisTable, aes(x = Course_Level , y = Counted_client)) + geom_boxplot() + theme_bw()+theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

multiplot(pairbox1, pairbox2, pairbox3, cols=3)

RELETIONSHIP BETWEEN TARGET FEATURE AND CATEGORICAL VARIABLE

CASE 1: TARGET FEATURE = Counted_clients

Box plot

pairbox4 <- ggplot(AnalysisTable, aes(x = Binned_Capacity, y =Average_clients)) + geom_boxplot()+ theme_bw()+theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

pairbox5 <- ggplot(AnalysisTable, aes(x = Binned_Capacity, y = Max_clients)) + geom_boxplot() + theme_bw()+theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

pairbox6 <- ggplot(AnalysisTable, aes(x =Binned_Capacity, y = Time)) + geom_boxplot() + theme_bw()+theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

multiplot(pairbox4, pairbox5, pairbox6, cols=3)

CASE 2: TARGET FEATURE IS CATEGORICAL

barpair1 <- ggplot(AnalysisTable, aes(x = Binned_Percentage, fill = Room)) + geom_bar(position = “dodge”)+ scale_fill_manual(values=c(“cyan4”,“yellow”,“orange”))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

barpair2 <- ggplot(AnalysisTable, aes(x = Binned_Percentage, fill = Binned_Time)) + geom_bar(position = “dodge”)+ scale_fill_manual(values=c(“blue”, “cyan4”,“yellow”,“orange”))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

barpair3 <- ggplot(AnalysisTable, aes(x = Binned_Percentage, fill = Course_Level)) + geom_bar(position = “dodge”)+ scale_fill_manual(values=c(“darkblue”,“blue”,“cyan4”,“yellow”,“orange”))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

multiplot(barpair1, barpair2, barpair3, cols=3)

ggplot(cabbage_exp, aes(x=Date, y=Weight, fill=Cultivar)) + geom_bar(position='dodge', stat='identity')

ggplot(AnalysisTable, aes(x = Binned_Percentage, y =Counted_client, fill = Room)) + geom_bar(position = "dodge", stat='identity')+ scale_fill_manual(values=c("cyan4","yellow","orange"))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

occupancy.lm = lm(Counted_client ~ Max_clients + Room + Binned_Time + Course_Level + Course_Level * Max_clients + Binned_Time*Course_Level, data=AnalysisTable) summary(occupancy.lm) plot(occupancy.lm)

plot(occupancy.lm, which = c(1), col = 1, add.smooth = FALSE, caption)

plot(AnalysisTable$Max_clients, resid(occupancy.lm), xlab = "Max_clients", ylab = "Residuals")plot(AnalysisTableRoom, resid(occupancy.lm), xlab = "Room", ylab = "Residuals") plot(AnalysisTable$Binned_Time, resid(occupancy.lm), xlab = "Binned_Time", ylab = "Residuals")plot(AnalysisTableCourse_Level, resid(occupancy.lm), xlab = "Course_Level", ylab = "Residuals")

Trellis

qplot(Average_clients, Counted_client, data = AnalysisTable, facets = . ~ Room) + geom_smooth(method=lm, fill="orangered3", color="orangered3")

qplot(Average_clients, Counted_client, data = AnalysisTable, facets = . ~ Binned_Time) + geom_smooth(method=lm, fill="orangered3", color="orangered3")

qplot(Average_clients, Counted_client, data = AnalysisTable, facets = . ~ Course_Level) + geom_smooth(method=lm, fill="orangered3", color="orangered3")

qplot(Max_clients, Counted_client, data = AnalysisTable, facets = . ~ Room) + geom_smooth(method=lm, fill="orangered3", color="orangered3")

qplot(Max_clients, Counted_client, data = AnalysisTable, facets = . ~ Binned_Time) + geom_smooth(method=lm, fill="orangered3", color="orangered3")

qplot(Max_clients, Counted_client, data = AnalysisTable, facets = . ~ Course_Level) + geom_smooth(method=lm, fill="orangered3", color="orangered3")

Bar plots

ggplot(AnalysisTable, aes(x = Binned_Time, y = Counted_client, fill = factor(Room))) + geom_bar(position = "dodge", stat="identity")+ scale_fill_manual(values=c("darkblue","cyan4", "yellow"))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

ggplot(AnalysisTable, aes(x = Course_Level, y = Counted_client, fill = factor(Room))) + geom_bar(position = "dodge", stat="identity")+ scale_fill_manual(values=c("darkblue","cyan4", "yellow", "orange", "blue"))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

ggplot(AnalysisTable, aes(x = Binned_Time, y = Counted_client, fill = factor(Course_Level))) + geom_bar(position = "dodge", stat="identity")+ scale_fill_manual(values=c("darkblue","cyan4", "yellow", "orange", "blue"))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))

ggplot(AnalysisTable, aes(x = Binned_Time, y = Average_clients, fill = factor(Room))) + geom_bar(position = “dodge”, stat=“identity”)+ scale_fill_manual(values=c(“darkblue”,“cyan4”, “yellow”, “orange”))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

ggplot(AnalysisTable, aes(x = Binned_Time, y = Average_clients, fill = factor(Course_Level))) + geom_bar(position = “dodge”, stat=“identity”)+ scale_fill_manual(values=c(“darkblue”,“cyan4”, “yellow”, “orange”))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

ggplot(AnalysisTable, aes(x = Room, y = Average_clients, fill = factor(Course_Level))) + geom_bar(position = “dodge”, stat=“identity”)+ scale_fill_manual(values=c(“darkblue”,“cyan4”, “yellow”, “orange”, “blue”))+theme_bw()+theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = “black”))

Linear Regression

occupancy.lm = lm(Counted_client ~ Max_clients + Room + Binned_Time + Course_Level + Course_Level * Max_clients + Binned_Time * Course_Level, data=AnalysisTable) summary(occupancy.lm) plot(occupancy.lm)

occupancy.glm = lm(Counted_client ~ Max_clients + Room + Binned_Time + Course_Level + Course_Level * Max_clients + Binned_Time * Course_Level, data=AnalysisTable)

Glm with poisson distribution

occupancy.poisson1 = glm(Counted_client ~ Max_clients + Room + Binned_Time + Course_Level + Course_Level * Max_clients + Binned_Time * Course_Level, family = poisson, data=AnalysisTable) summary(occupancy.poisson1) plot(occupancy.poisson1)

occupancy.poisson2 = glm(Counted_client ~ Max_clients + Room + Binned_Time + Course_Level + Course_Level * Max_clients + Binned_Time * Course_Level, family = quasipoisson, data=AnalysisTable) summary(occupancy.poisson2) plot(occupancy.poisson2)

library(MASS) occupancy.negbinomial <- glm.nb(Counted_client ~ Max_clients + Room + Binned_Time + Course_Level + Course_Level * Max_clients + Binned_Time * Course_Level, link = “log”, data=AnalysisTable)

plot(occupancy.negbinomial) warnings()

Logistic Regression

library(nnet)

glm.fit=multinom(Binary_Percentage ~ Max_clients + Room + Binned_Time + Course_Level, family = binomial, data=AnalysisTable) summary(glm.fit) #Prediction predict(glm.fit, AnalysisTable, “probs”)

vf5 <- varExp(form =~ Max_clients) M.gls5 <- gls(Counted_client ~ Max_clients + Room + Binned_Time + Course_Level + Course_Level * Max_clients + Binned_Time*Course_Level, weights = vf5, data=AnalysisTable)

vf1<- varFixed(~ Max_clients) M.gls <- gls(Counted_client ~ Max_clients + Room + Course_Level , weights = vf1, data=AnalysisTable)